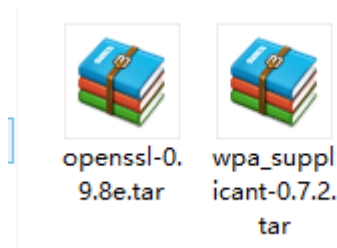


# Wifi 加密工具移植 (FS\_4412)

**WPA** 全名为 **Wi-Fi Protected Access**，有 WPA 和 WPA2 两个标准，是一种保护无线电脑网络(Wi-Fi)安全的系统，它是应研究者在前一代的系统有线等效加密 (WEP) 中找到的几个严重的弱点而产生的。

**OpenSSL** 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及 SSL。**SSL** 是 **Secure Sockets Layer** (安全套接层协议) 的缩写，可以在 Internet 上提供秘密性传输。

协议当我们移植好 USB-Wifi 驱动后，还需要移植 wpa\_supplicant，移植所需要的源码在 **Cortex-A9\移植部分\Wifi\wifi 加密源码** 目录中



## 1.Openssl 移植

将 **Cortex-A9\移植部分\Wifi\wifi 加密源码** 目录中 **openssl-0.9.8e.tar** 和 **wpa\_supplicant-0.7.2.tar** 拷贝到 Ubuntu 工作目录 (此文档以 **/home/linux/work/4412** 为工作目录)  
执行以下命令解压源码

```
tar xvf wpa_supplicant-0.7.2.tar.gz
tar xvf openssl-0.9.8e.tar.gz
```

解压完后可以看到以下两个目录

**openssl-0.9.8e** 和 **wpa\_supplicant-0.7.2**

```
linux@ubuntu:~/work/4412$ ls
DPO_MT7601U_LinuxSTA_3.0.0.4_20130913  toolchain-4.4.6
linux-3.0-fs4412_V3                    transplant
linux-3.0-fs4412_v3.tar                uboot-fs4412_v2.tar.xz
openssl-0.9.8e                         USB-Wifi driver.bz2
openssl-0.9.8e.tar.gz                  wpa_supplicant-0.7.2
project                                wpa_supplicant-0.7.2.tar.gz
```

### 1.1 安装补丁

openssl 源码编译需要打 wpa\_supplicant 提供的补丁，我们选择的 openssl 版本为 openssl-0.9.8e 所以我们需要 **wpa\_supplicant-0.7.2/patches/openssl-0.9.8e-tls-extensions.patch** 拷贝到 openssl-0.9.8e 源码目录下。  
执行命令

```
cp wpa_supplicant-0.7.2/patches/openssl-0.9.8e-tls-extensions.patch ./openssl-0.9.8e
```

进入 **openssl-0.9.8** 目录开始打补丁，执行以下命令

```
cd openssl-0.9.8e
patch -p1 < openssl-0.9.8e-tls-extensions.patch
```

## 1.2 配置编译安装

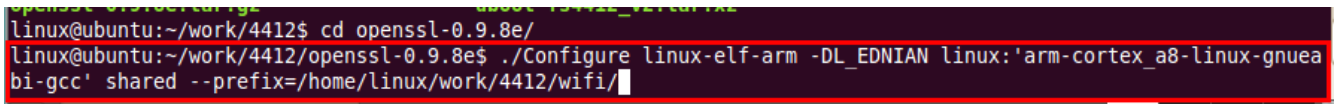
在工作目录 `/home/linux/work/4412` 里创建一个新的目录 `wifi`

```
mkdir /home/linux/work/4412/wifi
```

配置相关选项，在 `openssl-0.9.8` 目录中执行以下命令

```
./Configure linux-elf-arm -DL_ENDIAN linux:'arm-cortex_a8-linux-gnueabi-gcc' shared  
--prefix=/home/linux/work/4412/wifi/
```

如图所示

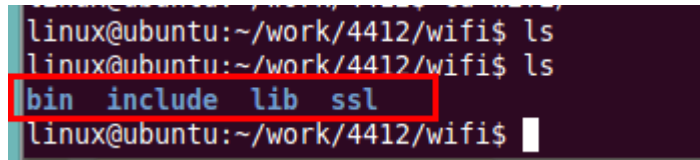


```
linux@ubuntu:~/work/4412$ cd openssl-0.9.8e/  
linux@ubuntu:~/work/4412/openssl-0.9.8e$ ./Configure linux-elf-arm -DL_ENDIAN linux:'arm-cortex_a8-linux-gnueabi-gcc' shared --prefix=/home/linux/work/4412/wifi/
```

配置命令执行后，开始编译安装，执行以下命令

```
make  
make install
```

以上步骤成功后，可以在 `wifi` 目录中看到如图所示的新增目录



```
linux@ubuntu:~/work/4412/wifi$ ls  
linux@ubuntu:~/work/4412/wifi$ ls  
bin include lib ssl  
linux@ubuntu:~/work/4412/wifi$
```

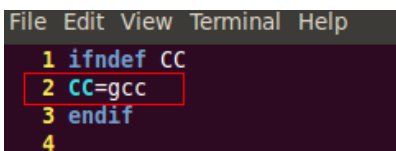
## 2.移植 wpa\_supplicant

进入 `wpa_supplicant-0.7.2/wpa_supplicant` 目录

```
cd /home/linux/work/4412/wpa_supplicant-0.7.2/wpa_supplicant
```

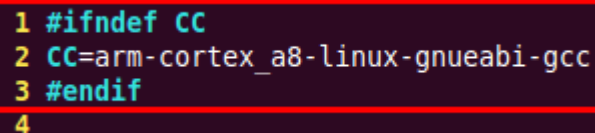
### 2.1 修改 Makefile

- 修改第 2 行中的 `CC=gcc` 为 `CC=arm-cortex_a8-linux-gnueabi-gcc`,并且**注销**第 1 行和第 3 行  
如下图所示



```
File Edit View Terminal Help  
1 ifndef CC  
2 CC=gcc  
3 endif  
4
```

修改为



```
1 #ifndef CC  
2 CC=arm-cortex_a8-linux-gnueabi-gcc  
3 #endif  
4
```

- 在

```
CFLAGS += -I../src  
CFLAGS += -I../src/utls
```

下添加:

```
CFLAGS += -I/home/linux/work/4412/wifi/include/
```

- 修改

```
LIBS += -lssl
```

为

```
LIBS += -lssl -L/home/linux/work/4412/wifi/lib/
```

➤ 修改

```
LIBS_p += -lcrypto
```

为

```
LIBS_p += -lcrypto -L/home/linux/work/4412/wifi/lib/
```

## 2.2 编译

在目录/home/linux/work/4412/wpa\_supplicant-0.7.2/wpa\_supplicant 下执行命令

```
cp defconfig .config  
make
```

编译完后可以再目录中发现以下两个文件 `wpa_supplicant` 和 `wpa_cli`

```
2/wpa_supplicant$ ls  
ents.c          preauth_test.c    wpa_gui-qt4  
ents.d          README            wpa_passphrase  
ents.o          README-Windows.txt wpa_passphrase.c  
amples          README-WPS        wpa_passphrase.d  
ss_rsn.c        scan.c            wpa_passphrase.o  
ss_rsn.h        scan.d            wpa_priv.c  
in.c            scan.h            wpas_glue.c  
in.d            scan.o            wpas_glue.d  
in_none.c       sme.c             wpas_glue.h  
in.o            sme.h             wpas_glue.o  
in_symbian.cpp  symbian           wpa_supplicant  
in_winmain.c    tests             wpa_supplicant.c  
in_winsvc.c     todo.txt          wpa_supplicant.conf  
kefile          vs2005            wpa_supplicant.d  
me.c            win_example.reg   wpa_supplicant.i.h  
me.h            win_if_list.c     wpa_supplicant.nsi  
ake.mak         wpa_cli           wpa_supplicant.o  
tify.c          wpa_cli.c         wps_supplicant.c  
tify.d          wpa_cli.d         wps_supplicant.h  
tify.h          wpa_cli.o         xcode  
tify.o          wpa_gui
```

➤ 将 `wpa_supplicant` 和 `wpa_cli` 拷贝到开发板根文件系统的 `/bin` 目录中  
(注意: 此文件系统是开发板的根文件系统)

## 3.测试

### 3.1 添加文件 `resolv.conf`

在开发板根文件系统的 `/etc` 目录中添加文件 `resolv.conf`  
其内容如下

```
# Generated by NetworkManager  
nameserver 202.96.64.68
```

### 3.2 添加文件 `wpa-psk-tnkip.conf`

在开发板根文件系统的 `/etc` 目录中添加配置文件 `wpa-psk-tnkip.conf`  
内容如下

```
# WPA-PSK/TKIP
ctrl_interface=/var/run/wpa_supplicant

network={
    ssid="XXXX"           # XXXX 为无线路由的 ssid
    key_mgmt=WPA-PSK
    proto=WPA
    pairwise=CCMP
    group=CCMP
    psk="XXXXXXXX"       # XXXXXXXX 为对应的无线路由密码
}
```

### 3.3 创建目录/var/run/wpa\_supplicant

在开发板根文件系统中创建目录/var/run/wpa\_supplicant，执行以下命令

```
mkdir /var/run/wpa_supplicant -p
```

### 3.4 启动开发板测试

等开发板内核启动完成后，在终端上执行以下命令

**注意：**以下步骤都是在 *USB-wifi* 网卡驱动安装成功后才能正常测试，*USB-Wifi* 驱动安装请参考《FS\_4412 平台 *USB-Wifi* 驱动移植参考文档》文档

- 配置网卡 ip,执行以下命令

```
ifconfig wlan0 192.168.1.200 //注意：配置的 ip 网段要和路由器的网段一致
```

- 配置默认网关

```
route add default gw 192.168.1.1
```

- 启动 wpa\_supplicant 连接无线网络

```
wpa_supplicant -B -i wlan0 -c /etc/wpa-psk-tkip.conf
```

- 查看连接状态

```
wpa_cli status
```